

A HELICOPTER FLIGHT PERFORMANCE
SYSTEM USING AN LSI MICROPROCESSOR

Edwin Eugene Elo

LIBRARY
MARIAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

A HELICOPTER FLIGHT PERFORMANCE
SYSTEM USING AN LSI MICROPROCESSOR

by

Edwin Eugene Eloë
and
Richard Tazewell Scott, Jr.

Thesis Advisor:

Gary A. Kildall

June 1973

T155173

Approved for public release; distribution unlimited.

A Helicopter Flight Performance System
Using an LSI Microprocessor

by

Edwin Eugene Elo
Captain, United States Marine Corps
B.S., University of Utah, 1966

and

Richard Tazewell Scott, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1973

ABSTRACT

This thesis presents the development of a helicopter gross weight calculator. Required qualities for aircraft system components such as durability, reliability and low weight are met using an LSI micro-processor. The prototype system which was developed weighs approximately four pounds and has approximate dimensions of 7" x 5" x 1/2". The cost estimate for the system is less than \$350. The calculator solution is based on the solution technique currently being used by Naval Aviators which is obtained from nomographs in the aircraft NATOPS Manual. Minor modifications are required to make this system applicable to different helicopter types. A listing of the calculator program and a discussion of the prototype's operation are included.

TABLE OF CONTENTS

I.	INTRODUCTION -----	6
II.	ALTERNATE SOLUTIONS -----	11
III.	SOLUTION -----	14
	A. HARDWARE -----	16
	B. DATA STRUCTURE -----	18
	C. SUBROUTINES -----	21
	D. MAIN PROGRAM -----	30
	E. SUMMARY -----	34
IV.	CONCLUSION -----	37
	APPENDIX A Operation of the prototype -----	38
	APPENDIX B Aircraft HOGE charts -----	40
	COMPUTER PROGRAM -----	43
	LIST OF REFERENCES -----	56
	INITIAL DISTRIBUTION LIST -----	57
	FORM DD 1473 -----	58

LIST OF FIGURES

1.	EXAMPLE SOLUTION USING HOGE CHART -----	9
2.	SCHEMATIC DIAGRAM OF THE PROTOTYPE INPUT OUTPUT ----	20
3.	PRESSURE ALTITUDE SECTION OF HOGE CHART -----	31
4.	HEADWIND SECTION OF HOGE CHART -----	33
5.	HOGE CHART FROM WHICH DATA WAS TAKEN -----	35

ACKNOWLEDGEMENTS

We wish to express our sincere appreciation to Ray Brubaker, who spent so much time and effort building the hardware interface equipment between the keyboard and the display LED's.

The authors are deeply grateful to Ken Farrell of USAATA for his assistance in explaining the alternate solution being tested by the Army and Air Force.

Last, but not least, we would like to thank Gary Kildall, our thesis adviser, for his advice and guidance during the course of our work.

I. INTRODUCTION

Each type of operational helicopter employed by the Navy and Marine Corps has an associated Naval Air Training and Operating Procedures Standardization (NATOPS) Manual. One section of the NATOPS Manual is comprised of a set of charts which may be used by aviators to determine specific values to measure the aircraft flight performance capabilities. In particular, one chart provides the aviator with a value for the maximum allowable gross weight for which the helicopter may be transitioned from forward flight to a hover without the contribution to lift capability provided by an associated "ground cushion" (Hover Out of Ground Effect). The HOGE Charts of the NATOPS Manual are in fact the primary method presently available to Naval Aviators for determining a predicted gross weight to hover. This parameter is considered critical to helicopter commanders for all operational flights. A standard aircraft mission for any given day may dictate variations in aircraft configuration due to "payload" changes and fuel consumption. These variations are generally not conducive to prescheduling. Atmospheric conditions and altitude requirements are subject to significant changes during any twenty-four hour period. These factors all contribute to necessitate continual updates of lifting capability.

The HOGE charts are generally nomographs which represent a compilation of test data for a given aircraft type. Input parameters to the nomograph are pressure altitude, ambient

air temperature and wind speed. To determine the gross weight hovering capability, the pilot enters the associated nomograph with a value for pressure altitude. The intersection of a line horizontal from the pressure altitude with the ambient temperature line projected vertically downward to the gross weight scale yields a value for the no-wind gross weight. Determination of the temperature line, unless it coincides with a range boundary, requires an interpolation process. The standard procedure for this process is to assume that all temperature lines are parallel to their closest boundary temperature line and that the distance between each successive temperature line is equal. The no-wind gross weight may be adjusted to account for wind speed. The gross weight determined above is used as an input to the second part of the nomograph. The graph is entered vertically with the no-wind gross weight value. This point is moved parallel to its closest wind line until it reaches a line corresponding to the input wind speed magnitude. The vertical projection of this intersection to the weight scale is the maximum gross weight with which a helicopter can hover under the given atmospheric conditions.

A sample solution is shown in Figure 1. At point 1, the pilot enters the nomograph with a pressure altitude of 10,000 feet. He then moves horizontally to point 2, the ambient temperature line of 15 degrees. Projecting point 2 vertically to point 2b produces the no-wind gross weight of 7000 pounds. To correct for a wind speed of twenty knots, the graph is

entered at point 2a with the no-wind gross weight value. This point is moved parallel to its closest wind line as shown to point 3 where the wind line intersects the headwind line associated with twenty knots. The vertical projection of point 3 to point 3a produces a gross weight value of 7875 pounds.

Before any landing or take off is made in a Naval helicopter, the aircraft commander is expected to insure, using a nomograph in the NATOPS Manual, that the maneuver is executable with respect to aircraft weight limitations. Assuming he knows his present gross weight, he subtracts it from the computed gross weight to determine if he can land. If the number is negative, he should not land. If it is zero, he can land but cannot increase the gross weight before taking off again. If it is a positive number, that is how much of a load he can safely take on.

Many times what actually happens is that either the pilot does not have a NATOPS Manual in the aircraft or he doesn't use it. It is much easier to rely on experience and guessing than to read the graph. The size of the NATOPS Manual tends to minimize its use in an aircraft cockpit. Also, some interpolation of the HOGE charts is required. Without a certain degree of care in the interpolation process, the resulting critical parameter will contain significant error. Aircraft vibration plus cockpit responsibilities are factors which may restrict effective chart interpolation in a cockpit environment. The pilot,

SAMPLE SOLUTION

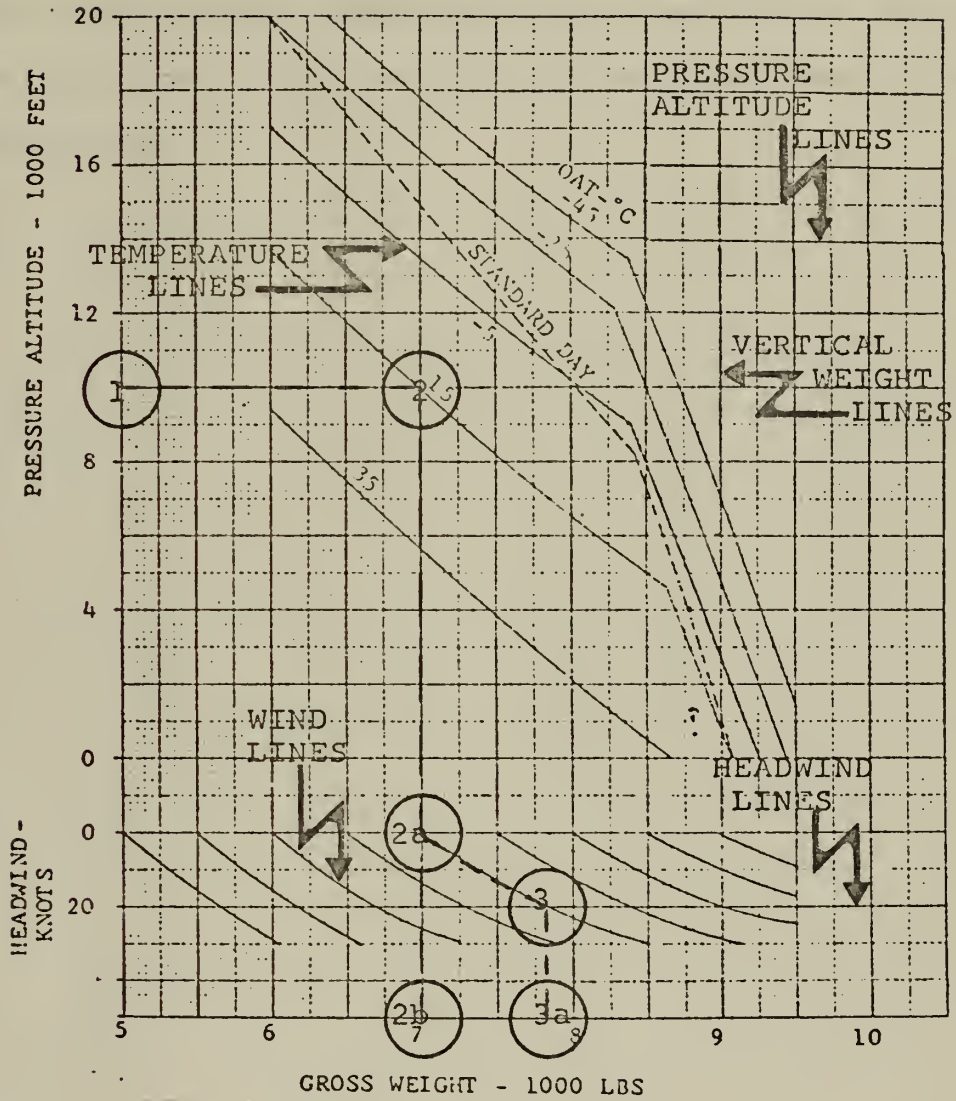


FIGURE 1

therefore, may attempt to land or take off when the graph should have told him it would be impossible. The Navy Safety Center states that 10% of helicopter crashes are caused by overweight conditions and they suspect that the figure could be as high as 40% if the actual causes could be determined.

II. ALTERNATE SOLUTIONS

In light of the previous discussion, it is evident that an improved technique for determining the maximum gross weight which will still permit a hover must be developed for cockpit use.

Under the Joint Army-Navy Aircraft Instrument Research (JANAIR) sponsorship, Transonics Incorporated has developed a Helicopter Lift Margin System (HLMS) described in reference 1, which is intended to fill this cockpit requirement. Flight tests are currently being conducted by the United States Army Aviation Systems Test Activity, USAASTA, at Edwards Air Force Base, to evaluate the effectiveness of Transonic's HLMS. It is claimed that this system will compute the gross weight of the aircraft using analog input from the engine and rotors. The engine inputs will tell how efficiently the aircraft engine is performing. Rotor input will tell how efficiently the rotor system is lifting the aircraft depending on the outside air temperature and air density at that time. The pilot would input headwind and altitude. Once the gross weight is computed, the theoretical maximum gross weight computed for the given conditions is computed. The actual gross weight is then subtracted from the theoretical maximum gross weight. The result is the lift margin for that specific aircraft on that specific time of day with those specific atmospheric conditions.

Another system developed by Marconi Elliott Avionics Systems Ltd. of Great Britain provides the pilot with a power margin indication in terms of a ratio between the present engine power output to maximum power output. The Elliott Helicopter Energy and Rotor Management System, HERMES, of reference 2, expects pilot input of temperature, static air pressure and helicopter gross weight. The temperature and pressure are used to compute engine power output based on data from Briston-Siddeley Technical Specification Number TSC 160 applicable to the Gnome H 1400. Using computed power output, the maximum weight lifting capability is computed. This value is then compared with the weight input to determine the power margin. The HERMES system is compact and weighs approximately eighteen pounds. The accuracy and reliability of this system as well as the general use potential and cost are not yet available.

The advent of micro-computers with their many favorable qualities such as low cost, high durability, low power requirements, miniature size and light weight, along with the fact that they are programmable, has produced a need to examine the potential of these computers in systems requiring these properties.

The fact that the previously mentioned systems have been developed is sufficient to justify the obviousness of the need. Caution should be exercised with the hardware selection to insure that the operational computer will best meet the system requirements.

The intent of this thesis is to investigate the suitability of the use of a micro-computer as an alternative method of solution to system requirements of the type described. For this investigation, the primary considerations are the following: the system must possess an acceptable degree of accuracy, it must produce usable output, it must be adaptable to a helicopter cockpit, and it must be competitively priced. The application discussed here uses an MCS-4 micro-computer built by Intel, described in Reference 3, as the basis for the design of a prototype system which will accept a set of input parameters and produce the gross weight capabilities of a helicopter as output.

III. SOLUTION

The Nomographs which are presently used by Naval Aviators to provide aircraft performance limits were divided into regions as indicated in Figure 3. The curve segments within each region are represented in the solution by linear approximations of those segments using the end points of each segment to determine the respective slope and intercept. The slope of a given segment is not necessarily equal to the slope of the preceding segment in that region; therefore, slopes are adjusted using the differential slope between two segments in a region and the associated differential temperature. For this application, the temperature differential between any two segments is assumed to be linear. The computed values of slope and intercept for each segment within each region are stored in a data table.

The input temperature dictates the starting position in the data table. A higher likelihood exists that the computed value of gross weight will fall in region II or region III based on standard day temperature and normal altitude requirements. For this application, the first computation uses coefficients associated with segments in region II rather than region III. This eliminates the possibility of three iterations to compute the no-wind gross weight. Once the initial value for gross weight is determined, the magnitude is compared to a test value which separates region II from region III. In the event that the value for gross

weight is less than the test value, a second test is made to determine if the gross weight is in region I. If either test condition is met, the coefficients for that region are used to compute a new no-wind gross weight. If neither test condition is met, the computed value is the no-wind gross weight.

The value of no-wind gross weight is used to determine the entry point into the wind speed correction data table. The seven wind speed regions are each segmented at intervals representing ten knots of wind speed. Coefficients corresponding to segment slopes are loaded from the data table to adjust the gross weight for wind. The wind speed adjustment is made in increments of ten knots. The coefficient is updated after each computation and input wind speed is reduced by ten knots. The final adjustment for wind speed is made when the current value of wind speed is between zero and ten knots. The resultant is output as the maximum gross weight for HOGE relative to the set of input parameters. A device which will accept the indicated input parameters, operate on those parameters, and output a value for the maximum gross weight was constructed. The device hereafter referred to as the Helicopter Gross Weight Indicator, HGWI, employs components available to students at the Naval Postgraduate School micro-computer laboratory. The hardware for the HGWI prototype consists of an MCS-4 computer interfaced with an input-output device. The software is comprised of a data structure, a set of sub-routines and a main program.

The data structure is the section of software which is aircraft type dependent. It consists of a table or set of tables in read-only-memory (ROM) containing constants extracted from a specific nomograph. The HH-IK helicopter was arbitrarily selected as the representative aircraft for development of the data structure in this particular HGWI. Further references to this particular system use the designation HGWI/HH-IK.

The main program is an algorithm which, using a set of subroutines and the data structure, operates on the input parameters to compute a solution for output by the HGWI. The hardware, data structure, subroutines and main program are further described in subsequent sections.

A. HARDWARE

The MCS-4 micro-computer set [Ref. 3] consists of the following three chips, each packaged in a sixteen pin DIP package:

- (1) A Central Processor Unit Chip (CPU 4004)
- (2) A Read Only Memory Chip (ROM 4001)
- (3) A Random Access Memory Chip (RAM 4002)

The prototype also includes a keyboard type input-output device with its associated electronics. This keyboard in the production model would have to be modified to eliminate the illegal keys and excess Light Emitting Diodes.

1. 4004 CPU

The CPU contains the control unit and the arithmetic unit. It works in conjunction with the 4001 ROM'S and the

4002 RAM'S and the I/O package to form a completely self contained system. The CPU communicates with the RAM'S and ROM'S of the system through a four line data bus and with the I/O package through ROM ports.

2. 4001 ROM

The 4001 ROM is a 2048 Bit Metal Mask programmable ROM. Each chip is organized as 256 x 8 bit words which contain the program and data table. Each chip also has a four bit input-output (I/O) port which is used to route information to and from the data bus lines in and out of the system. The prototype uses eight 1702 programmable ROMS. The 1702 is equivalent to the 4001 ROM except that the 1702 may be erased and reused. Seven 4001 ROMS would be used in the production model.

3. 4002 RAM

The 4002 RAM stores three hundred twenty bits arranged as four registers of twenty-four bit characters which are divided into sixteen main memory characters plus four status characters. The prototype contains four 4002 RAM chips.

Input is accomplished by a standard mechanical type keyboard in the prototype. Output is through sixteen light emitting diode digits which are connected to the CPU through a shift register and a decoder/driver, as in Figure 2.

When a key on the keyboard is depressed, this mechanically sets up a BCD code at the output of the keyboard. This code contains one extra bit which tells whether the

depressed key is a number or a control key. If the key is a number, it is transferred, by software, to a designated position in RAM register fifteen. If it is a control key, program control is transferred to the section of the program which will correctly process the key. Since all keys on this particular keyboard are not required, all illegal keys are ignored. Repeatedly during the input process, a routine is called to display RAM register fifteen. This display routine sets one bit of the shift register. This bit is shifted at every clock pulse. As the bit is shifted, the corresponding LED is enabled. The number which is displayed on the enabled LED is the input to the decoder/driver. This input comes from the enabled position of RAM register fifteen. The enabled position of RAM register fifteen is controlled in the same manner as the enabled LED. Each position of RAM register fifteen will contain either a BCD zero through nine or a fifteen. BCD digits are displayed on the enabled LED as decimal digits, while a fifteen is displayed as a blank.

Because the shifting is so rapid, it appears to the user that the whole decimal number is being displayed at one time. In actuality, only one digit is being displayed at a given instant by the DISPLAY routine.

B. DATA STRUCTURE

Two tables of data are required for the HGWI/HH-IK. One table is a four by fifteen array which contains the constants required to compute the no wind gross weight.

There are four twenty-degree temperature ranges depicted in Figure 3, where temperature units are degrees centigrade. Range 1 is from -45° to -25° , range 2 is from -25° to -5° , range 3 is from -5° to 15° and range 4 is from 15° to 35° . The columns of Table 1 correspond to these temperature ranges. Each temperature range is divided into three regions to improve the linearity assumption. Region I is separated from region II by the 7,500 pound gross weight line. Region II is separated from region III by a line connecting the points where the change in slope of the two lines describing a range is maximum. Therefore, each column of Table I contains the coefficients needed to describe four line segments; one within each region and one which separates region II and region III. The second table is a seven by three array. The gross weight for the HGWI/HH-IK may range from 6,000 to 9,500 pounds. The curves which describe the correction for wind speed extend from each 500 pound increment along the gross weight line. The seven increments correspond to the seven columns of Table II. Again, to improve accuracy, the curves of each increment are defined at ten knot intervals. The HGWI/HH-IK corrects for wind speeds up to thirty knots; therefore, there are three rows per column of Table II.

The constants stored in Table I represent the slopes, differential slopes, intercepts and differential intercepts of the line segments as described above. Table II consists of the slopes of the line segments used to correct for wind speed. All constants are stored in a single ROM

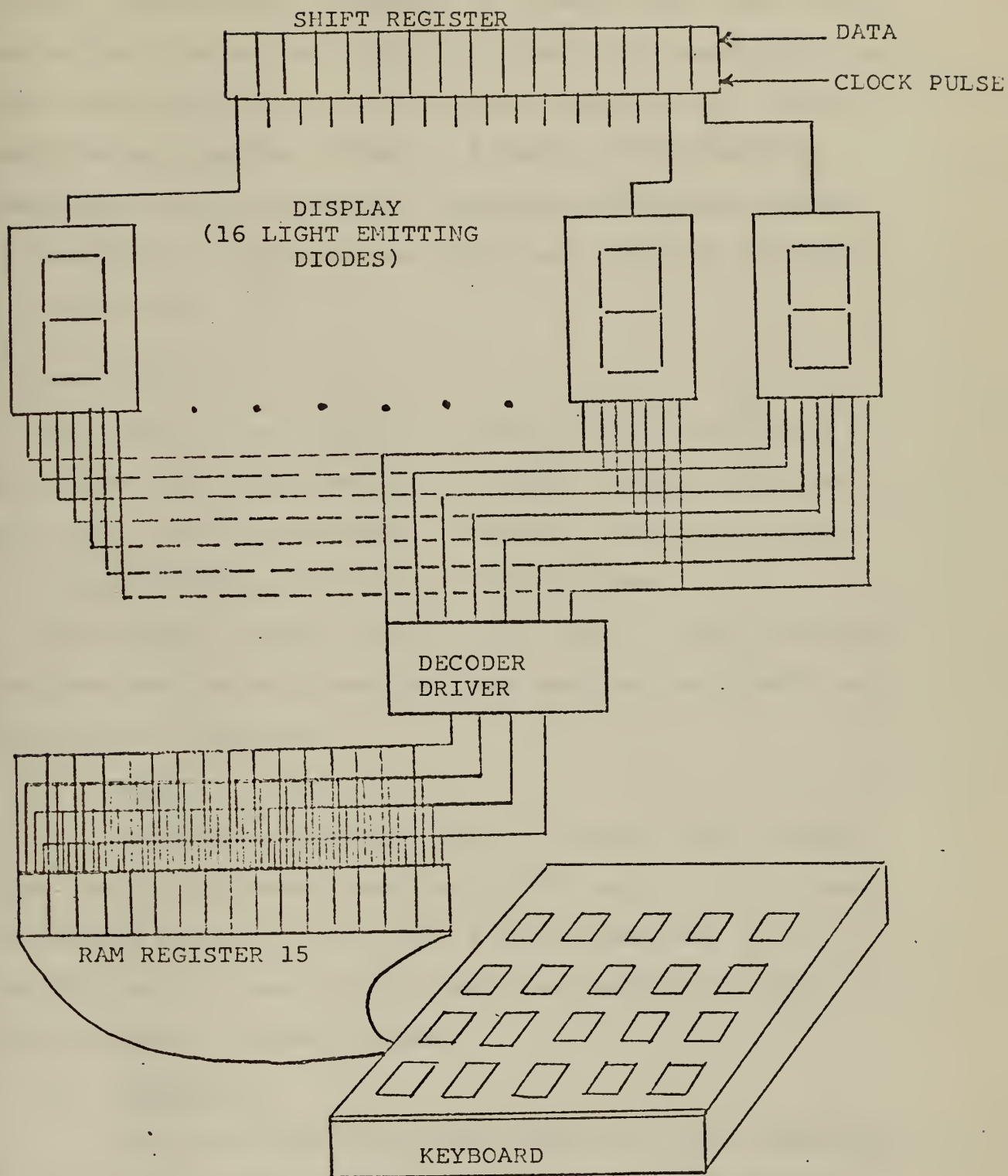


FIGURE 2.

chip. Two eight bit locations are required per constant stored. Each number is stored in reverse order since this technique is more compatible with the MCS-4 instruction set. The upper four bits of each constant are used to indicate positive or negative numbers. A zero in this location indicates positive numbers. Negative numbers are stored in the ROM in tens complement form and the sign character is set to one.

C. SUBROUTINES

The subroutines used in the HGWI/HH-IX Calculator can be divided into three main categories: general purpose, arithmetic, and input-output routines. Appendix C gives a complete listing of the calculator program.

The general purpose routines are used by the arithmetic routines and the main program, and perform tasks which are consistently repeated.

1. MOVE

The MOVE subroutine takes a sixteen digit number from the RAM register specified by CPU register pair ten-eleven and moves it along with status character zero to the RAM register and status character specified by CPU register pair fourteen-fifteen.

2. COMPLEMENT

The subroutine COMPLEMENT takes the Tens complement of the sixteen digit number which is in the RAM register specified by CPU register pair fourteen-fifteen.

3. SHIFLT

The SHIFLT routine takes the sixteen digit number which is in the RAM register specified by CPU register pair twelve-thirteen and shifts it left the number of decimal places specified by CPU register fifteen. The number in CPU register fourteen must agree with the number in CPU register twelve.

4. ZERORR

ZERORR takes the RAM register specified by CPU register pair fourteen-fifteen and places zeroes in each of its sixteen positions and sets its status characters to zero.

5. SHIFRT

SHIFRT is used to take a specified RAM register and shift it right a specified number of decimal digits. The programmer is responsible for making sure that the correct digits of the correct register are specified. SHIFRT assumes that CPU register pair twelve-thirteen contains the address of the rightmost digit of the desired RAM register. In addition, SHIFRT insures that CPU register pair fourteen-fifteen contains the address of the rightmost digit in the desired RAM register minus the number of places to be shifted.

6. SWITCH

SWITCH is a data manipulation routine which takes two eight bit words from the ROM address specified by CPU register pair zero-one. The first one-and-a-half words are the desired number in BCD. These three BCD digits are stored

in the first three positions of the specified RAM register. The last half of the second word in RAM is written into status character zero. If this is an odd number, it indicates the number is negative and this number is in tens complement form. As a result, nines must be placed in positions three to fifteen of the designated RAM register.

7. INIT

INIT initializes the CPU by placing zeroes in CPU registers two through fifteen, a one in register twelve, and a two in register fourteen. INIT also clears both the accumulator and the carry.

8. LOAD

Subroutine LOAD is required because of an idiosyncrasy of the machine. With the MCS-4 one cannot fetch indirect (FIN) from ROM unless the FIN instruction is on the same chip as the constant to be fetched. LOAD transfers an eight bit word from ROM to CPU registers two and three.

9. INITC

INITC is a routine which initializes the CPU registers with constants which are needed to compute the address of entry into a table of constants. These constants are particular to an aircraft type; therefore, they are stored at the base of data Table I.

10. ASCOL

After INITC has loaded into CPU register four the RAM location which contains the computed no-wind gross weight, ASCOL is called to inspect the contents of that RAM. The

column of Table II which contains the constants of interest is numbered corresponding to the number of 500 pound increments above a given base value. ASCOL sets a flag by incrementing CPU register eleven if the weight value in the designated RAM location has a hundreds digit of five or more. The value of the thousands digit is then loaded into CPU register nine. Control is then returned to the calling program.

11. COLNUM

When COLNUM is called, INITC will have loaded a base value in CPU register six, and either TEMPCOL or ASCOL will have loaded CPU register nine with a value to be compared to this base. The base value is the minimum value of temperature if access of Table I is desired or the minimum thousands digit on the gross weight scale if accessing Table II. COLNUM compares register nine with register six and increments the register six value until it is greater than the value in register nine. By fixing the value of the increment and counting the number of increments made, the column number of Table I or the number of thousand pound increments which will be used to determine the column number of Table II, is computed and stored in CPU register eight.

12. COLNUMAS

The subroutine COLNUMAS computes the column number of Table II. COLNUMAS doubles the value in CPU register eight which was returned from COLNUM and adds CPU register eleven which was set by ASCOL. This result is left in CPU register eight.

13. LOADROBAS

This routine loads from the table of constants the number of rows per column in a given table and stores the value in CPU register seven. The base address of the desired table of constants is loaded into CPU register pair zero-one.

14. ASCHECK

To insure that the input value of wind speed is acceptable, ASCHECK inspects the wind speed RAM location. Values of wind speed greater than thirty knots cause ASCHECK to initiate the error routine.

15. TEMPCOL

Prior to calling TEMPCOL, the RAM location of the scaled input temperature will be loaded into CPU register pair four and five by INITC. TEMPCOL checks the value in that location to insure the input temperature is greater than -45° . If not, the value is outside the range of this application and an error condition results. If the input temperature value is acceptable, TEMPCOL loads the tens digit of that value into CPU register nine and increments it by one if the value of the units digit is greater than zero. The number returned in CPU register nine will be referenced by COLNUM to determine the column number of Table I to access.

16. ADDRS

This subroutine is called after the CPU registers have been initialized. The column number of interest will

have been loaded into register eight by TEMPCOL and COLNUM for Table I or by ASCOL, COLNUM and COLNUMAS for Table II. The table base address was loaded into register pair zero-one and the value of the number of rows per column was loaded into register seven by LOADROBAS. ADDR5 then computes the address of entry into the respective table of constants by adding to the table base address the product of the column number and the number of rows per column. Control is returned to the calling program with the required address in register pair zero-one.

17. TESTPA

The input value of pressure altitude is inspected by TESTPA. Values greater than 20,000 result in an error condition.

The arithmetic routines are used in the computation of the problem. All negative numbers are in tens complement form with an odd number stored in status character zero. This status character is an easy means of detecting negative numbers. The procedure is to read the status character into the accumulator, clear the carry bit and then rotate the accumulator right one bit. If the carry is on, the number is negative; if it is not on, the number is positive.

18. ADDIT

An arithmetic routine which takes two sixteen digit numbers in RAM registers zero and one and adds them together. The result is in RAM register zero.

19. SUBTRACT

Subroutine SUBTRACT takes a sixteen digit number in RAM register one and subtracts a sixteen digit number in RAM register zero from it. If the number in RAM register zero is negative, the routine recomplements it and adds it to the number in RAM register one. The resulting number is always in RAM register zero.

20. MULT

MULT is a six digit multiply routine using the shift-and-add algorithm. MULT calls SETUP which determines whether the answer will be a positive or a negative number. This is determined by adding the status characters of the two numbers. If the result is an odd number, the multiplication will yield a negative number. If it is an even number, the resulting multiplication is positive. The result of the multiplication is in RAM register zero.

Input-output control of the program is accomplished through CHECKNOSTROBE. The program remains in a loop as long as there is no key on the keyboard depressed. Inside this loop is a call to subroutine DISPLAY which displays the sixteen digit contents of RAM register fifteen. When a key is not depressed, ROM port 1 is read as a one and the loop continues. As soon as a key is depressed, control of the program is transferred to CHECKSTROBE. CHECKSTROBE uses the five-bit key code to index a table called KEYTAB. KEYTAB consists of thirty two jump unconditional instructions each of which corresponds to a legal key or an illegal key.

There are twenty seven keys on the prototype, sixteen legal keys and eleven illegal keys. Control is transferred to the correct code for the depressed key through these jump unconditional instructions in KEYTAB. There are seven routines for legal keys. An illegal key merely transfers control back to CHECKNOSTROBE. The legal keys are described below.

1. Number

The code associated with a number key takes a decimal digit and stores it into RAM register fifteen in the next available digit position. Control is then transferred back to CHECKNOSTROBE.

2. Temperature

The code for the temperature key takes the decimal number in RAM register fifteen, adds forty-five to it as a scale factor and places the result in RAM register A. Control is transferred to OUTCLEAR which loads fifteen into RAM register fifteen. This character displays as a blank. OUTCLEAR then transfers control to CHECKNOSTROBE.

3. Speed

When the speed key is depressed, this code transfers the decimal digit of RAM register fifteen to RAM register six, then transfers control to OUTCLEAR which, in turn, transfers control to CHECKNOSTROBE.

4. Altitude

This key transfers the decimal digits in RAM register fifteen to RAM register five and, after shifting left two places, control is transferred back to CHECKNOSTROBE through OUTCLEAR.

5. Negative

This code allows negative temperatures to be placed into the prototype. If the number is negative when the temperature key is depressed, the contents of RAM register fifteen is complemented and then the forty-five scale factor is added.

6. Compute

The compute key transfers control to the start of the main program. After computation has been completed and the correct answer is stored in RAM register fifteen, control is returned to CHECKNOSTROBE which will display the result while waiting for further input.

7. Clear

When the clear key is depressed, this code transfers control to OUTCLEAR which blanks out RAM register fifteen and then returns control to CHECKNOSTROBE.

Two service routines are used by CHECKNOSTROBE and CHECKSTROBE. They are:

1. DELAY

The DELAY routine is a double nested loop which, due to instruction cycle time, results in a five hundred micro-second wait.

2. CHECK

CHECK is a service routine which, during each transfer of a number from RAM register fifteen to another register, checks to see if the number being transferred is a fifteen. If it is, the fifteen is not transferred and control is returned to where the subroutine CHECK was called.

D. MAIN PROGRAM

The main program is the problem flow controller for the HGWI. This section of software calls the available sub-routines to perform necessary arithmetic operations and character manipulations in order to arrive at a solution. The main program is entered with the input parameters loaded into their respective save registers of RAM. Register five contains the pressure altitude, register six holds the wind speed and register ten contains the scaled temperature. Registers four and nine are temporary locations used to save TI, a temperature increment used to adjust the base temperature line within a given range, and the various stages of computation of gross weight, respectively. The CPU registers are cleared and then initialized with constants which enable subsequent subroutines to compute the input temperature range. The column number, the number of rows per column, and the base address of Table I will permit computation of the address of entry into Table I. As was mentioned, the no wind gross weight for this application is determining by first computing a weight value by assuming the input parameters, temperature and pressure altitude, will fall in region II of Figure 3. The algorithm used in the solution of this application consists of the following steps:

1. First determine an initial value for W which satisfies the equation:

$$W = A(1-B \cdot TI)PA + C + D \cdot TI$$

Where A is the slope of the segment representing the highest

temperature in region II, B is the difference between the slopes of the bounding segments divided by the difference in temperature representation of the bounding segments, and C is the value of the intercept along the abscissa of the base segment within a particular region. The value D is the difference between the intercepts of the bounding segments of a particular region divided by the difference in temperature representation of those bounding segments.

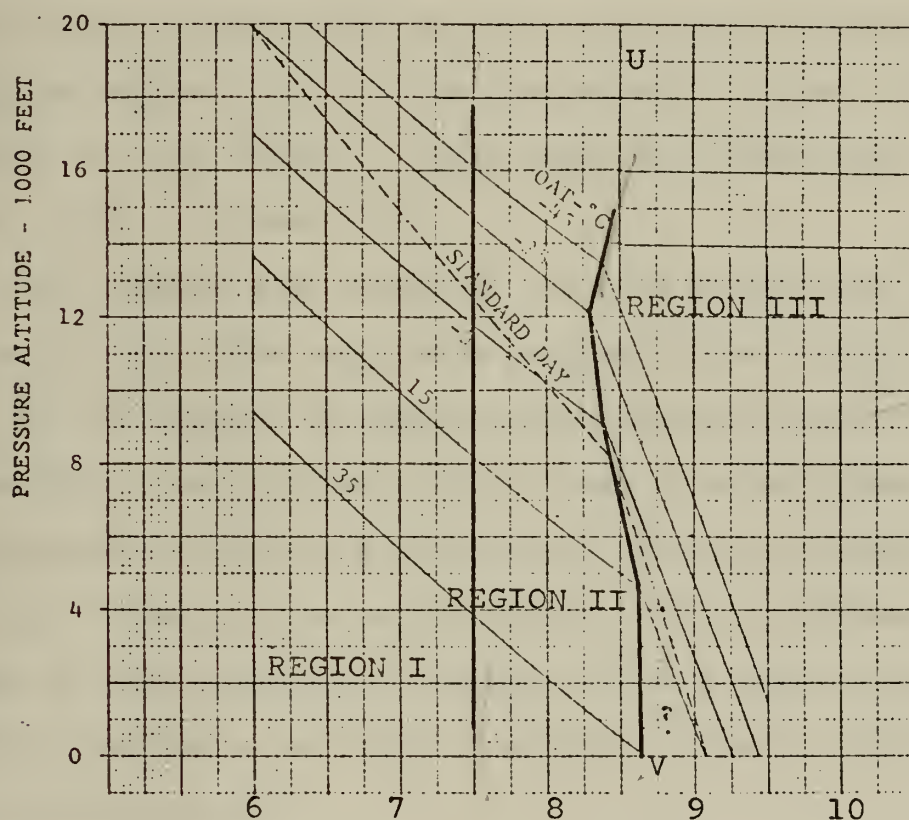


FIGURE 3 GROSS WEIGHT - 1000 LBS.

TI is the difference between the value of temperature represented by the base segment and the scaled input temperature, and PA is the input pressure altitude. The constants A, B, C and D are available in Table I. They are stored with at most three characters of significance.

2. The next step is to determine a value for W_t which satisfies the equation:

$$W_t = E \cdot PA + F$$

Where E is the slope of UV, the line which separates region II from region III, and F is the intercept along the abscissa of UV.

3. The computed value of W is then compared to the computed value of W_t .

If W is greater than W_t , then the input parameters must fall in region III. If the comparison of step 3 determines that W is less than W_t , then step 3a is executed. Otherwise, step 4 is executed.

3a. Compare the computed value of W with the weight value, 7500, which separates region I from region II.

If the compare of step 3a determines that W is greater than 7500, then W must be the final no-wind gross weight. If, however, W is less than 7500, step 4a is executed.

4. Finally, the new constants A3, B3, C3 and D3 applicable to the segments of region III are substituted for the initial values in equation one and a final no-wind gross weight is computed.

4a. Upon reaching this final step, new constants A1, B1, C1, and D1 are substituted into equation one. W, the final no-wind gross weight, is computed.

For the HGWI/HH-IK, the final no-wind gross weight computed above must be adjusted to compensate for the scale difference between the units along the ordinate and abscissa

of Figure 3. The main program beginning at the label XINIT accomplishes this adjustment with the equation:

$$W = W/4 + 6000$$

If a flag had been set to indicate the presence of wind, the no-wind gross weight is now modified to account for the contribution to lifting capability due to wind. The correction is made using the equation:

$$W = K \cdot AS + W_0$$

Where K is the scale factor, 50, times the slope of the segment of interest, AS is the magnitude of the input wind speeds, and W_0 is the previously computed no-wind gross weight.

The column number for entry into Table II is determined by inspecting W_0 . For this application, the slope of the weight correction lines is changed at each ten knot increment. Therefore, for values of wind speed in excess of ten knots, the first W computed replaces W_0 for the next iteration. This process, illustrated by dotted lines in Figure 4, may need to be repeated at most twice. The computed gross weight is then moved to the display register.

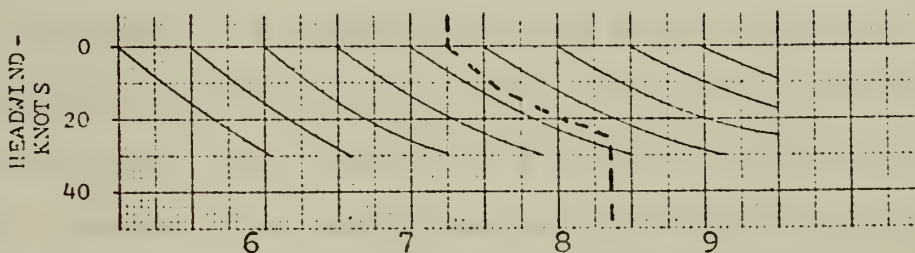


FIGURE 4 GROSS WEIGHT -- 1000 LBS.

E. SUMMARY

The data for this application was extracted from the nomograph shown in Figure 5. The information produced by this system is therefore of the same quality and value as that obtainable directly from the nomograph. The nomographs are constructed from data obtained from test flights and are adjusted to apply to a helicopter equipped with a specification engine. An adjustment is made to allow for a five per cent fuel gauge error. Manufacturers guarantee that all engines will perform at or above the level of a specification engine. The gross weight values are therefore maximum only to an aircraft with a specification engine.

Nomographs applicable to several models of operational helicopters are included in Appendix A. The similarity of the various nomographs would allow, with minor modification to this system, a general system applicable to any of the associated aircraft. The only requirement would be to replace the ROM chip which contains the tables of constants.

A system comparable to the prototype could be produced at an approximate unit cost of \$350. For an associated cost increase, the system could be easily adapted to analog inputs from temperature and pressure sensing devices, requiring only that the pilot dial in the headwind speed. In this manner, one could have a continuous read-out of maximum gross weight for hovering. This would be particularly useful for the loading of troops or cargo because before the pilot lands, it would be possible to know how much added weight he could take on.

MAXIMUM GROSS WEIGHT FOR HOVERING

OUT OF GROUND EFFECT - MILITARY POWER

3.5°C INLET TEMPERATURE RISE

Model(s): UH/TH-1L, HH-1K
Data as of: August 1969
DATA BASIS: Army Phase D

ENGINE RPM 6600

Engine(s): T53-L-13B
Fuel Grade: JP-4/JP-5
Fuel Density: 6.5 Lb/Gal.

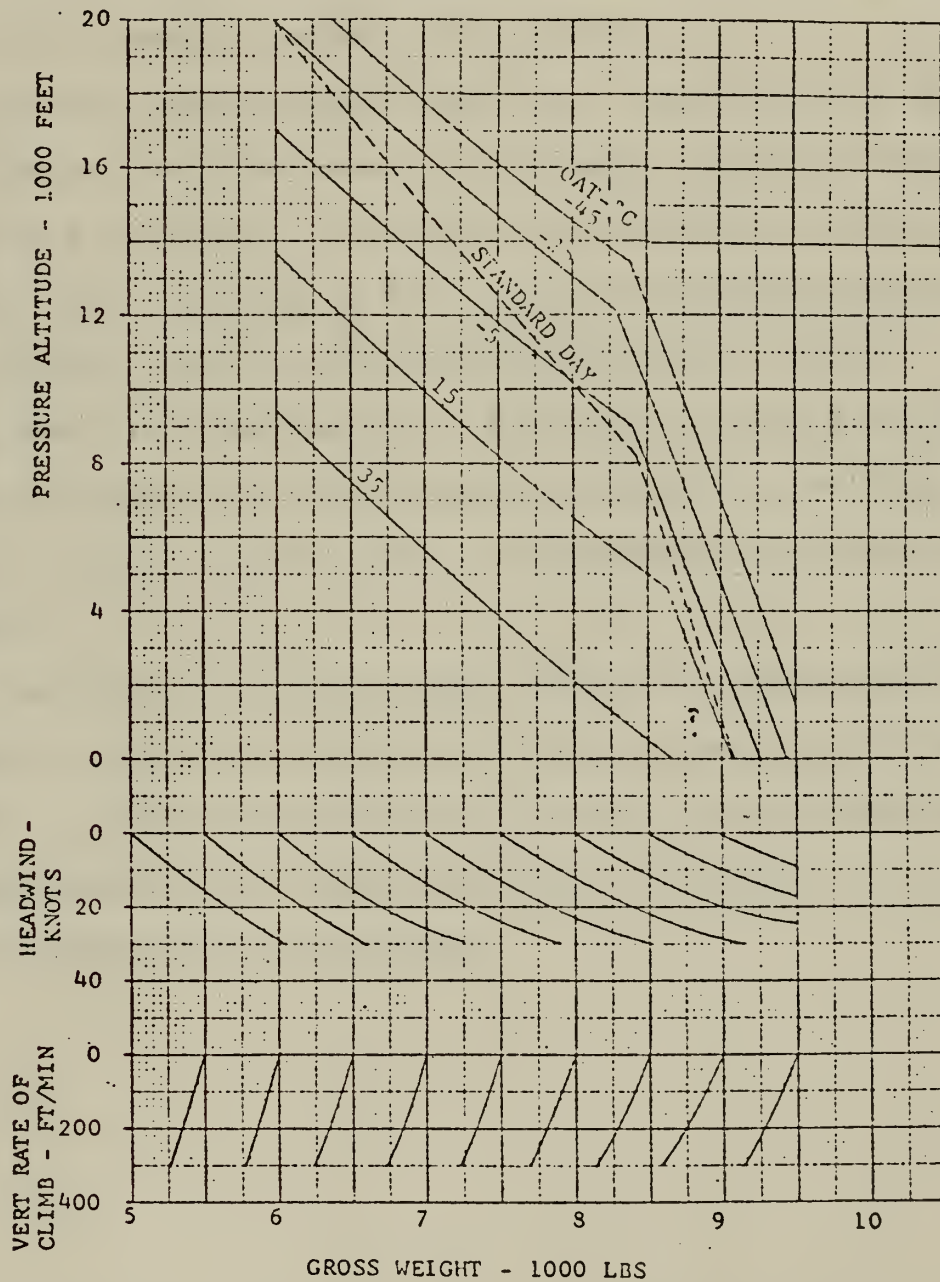


FIGURE 5.

If it were desired to allow the pilot to vary any or all of the inputs, it would be possible to keep the analog input, but also be able to set the variable at any desired value. This would allow a pilot to determine his safety margin before every take off and landing.

The primary contribution made by a system similar to the prototype described here is the rapid on-board availability of a measure of the hovering capability of the aircraft. The validity of this measure could be improved with a careful analysis and possible modification of existing data representations. Also, due to the fact that the micro-computer is programmable, alternative methods of solution could be tested until an acceptable technique is developed.

An operational system of this design is applicable to an aircraft model. A system which will produce performance data for a particular aircraft of a model is more desirable but significantly more expensive both for initial installation and subsequent maintenance.

IV. CONCLUSION

A low cost, accurate, effective prototype of a micro-computer-based helicopter flight performance system has been built. Improvements are needed in keyboard design, data input and data storage techniques, but the feasibility of making a helicopter flight performance system work with a microcomputer has been proven. Prior to this, it was known that a man, given the time, could perform the task of figuring out maximum gross weight for hovering. It was also known that a sophisticated, but expensive, computer system could be built to accomplish basically the same task as the pilot except a computer would prove more accurate and faster than an interpolated answer. Now it is known that it is possible to build a less expensive, less sophisticated computer system. This computer system would rely on the pilot to input the desired parameters, but would produce an accurate result for a fraction of the cost of the more sophisticated system.

APPENDIX A

Prototype operation requires depressing a sequence of number keys followed by a control key until all desired input parameter values are loaded. Parameter values for the HGWI/HH-IK must be within the ranges given below. The compute key may be depressed at any time. When the compute key is depressed, the values which are currently in the machine are used for the computation of maximum gross weight for hover.

To clear the values of temperature, altitude and wind speed, one merely depresses the corresponding control key. If one only desires to enter or change one number, all that is necessary is to enter the desired number and then depress the correct key. This will place the new number in the desired location and leave the other two inputs the same as before. The same procedure is followed if more than one number is to be changed.

There are certain ranges for acceptable input values. These are due to the ranges on the graphs and also limitations of the aircraft. If an input number is outside of these ranges, an error will be indicated by all nines appearing on the display. The acceptable ranges are:

TEMPERATURE: -45° C. to 35° C.

WINDSPEED: 0 Knots to 30 Knots

ALTITUDE: 0 Feet to 20,000 Feet

If an error is encountered, one merely depresses the EM

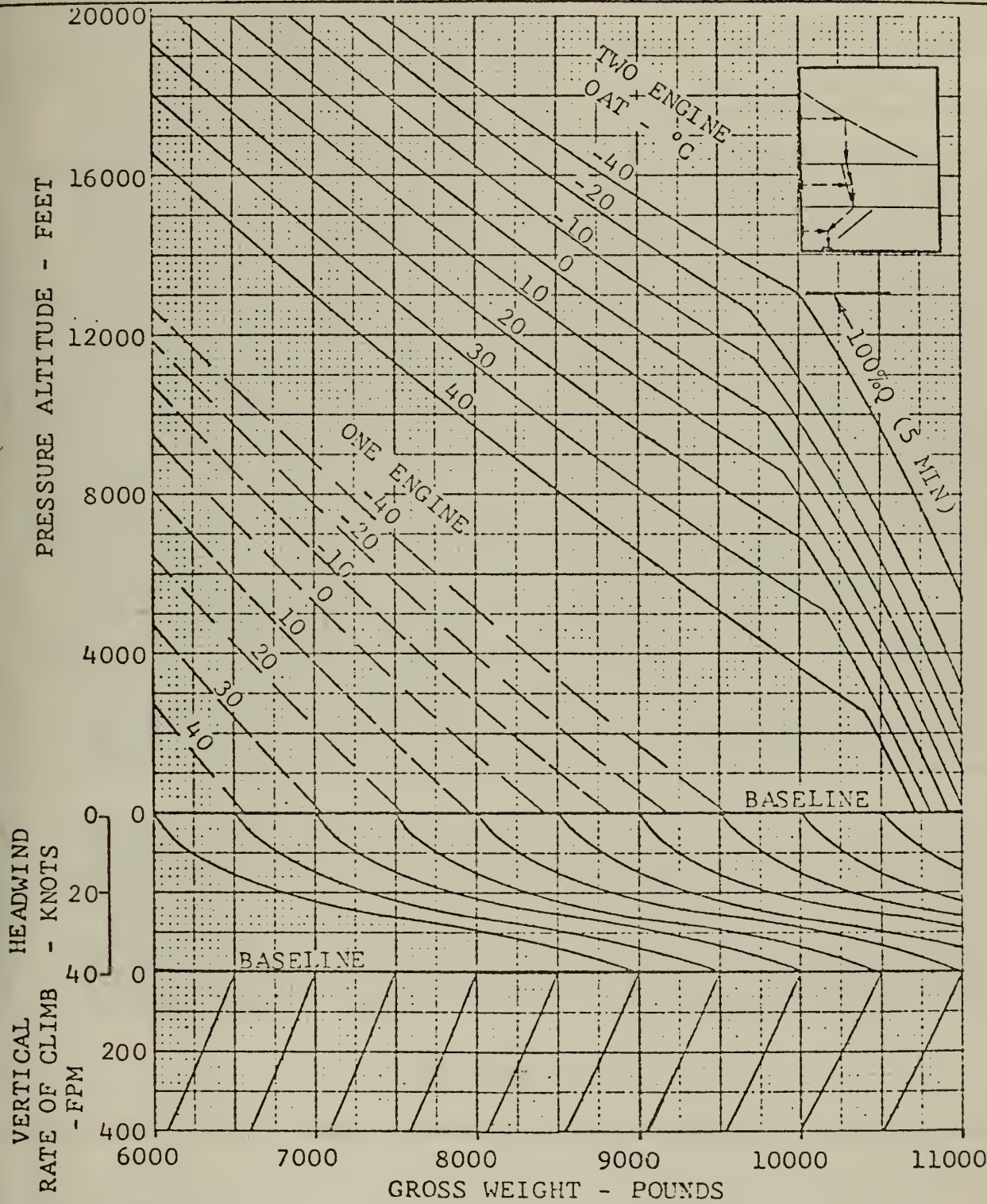
(clear) key and then inputs the correct number. If no error is encountered, the maximum gross weight for hovering will appear in the display after the compute key has been depressed. The pilot must take this number which is displayed and compare it to his present gross weight. From this comparison, the pilot decides whether a landing or take off can be made.

It is possible to recompute the maximum gross weight for hovering with different inputs with this prototype. The pilot merely depresses the clear key to clear the display and then enters the values of pressure altitude, temperature, and/or wind speed he wishes to change. After these are entered, he depresses the compute key and the new gross weight is displayed.

HOVER CEILINGS (OUT OF GROUND EFFECT) - MRP

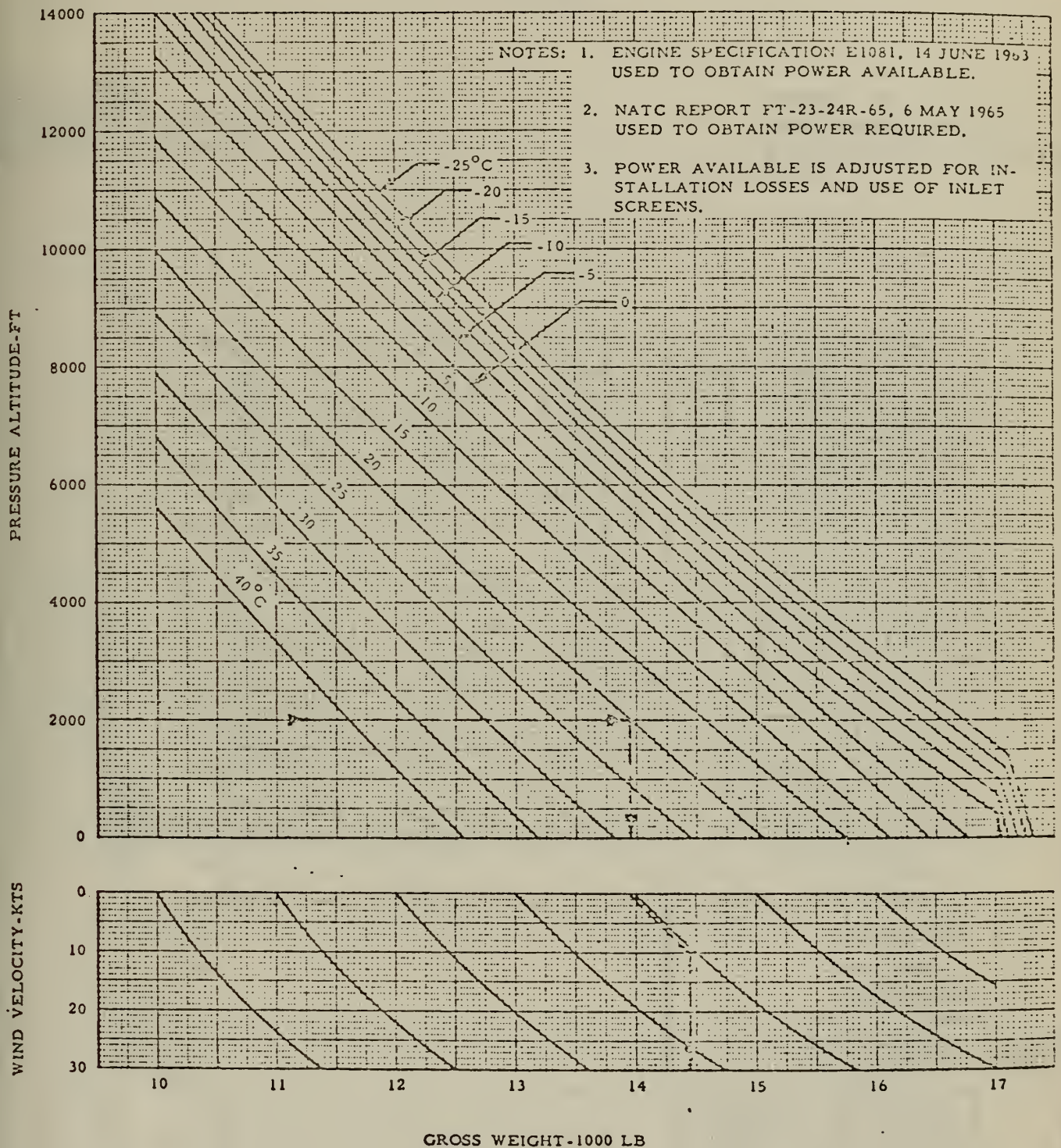
Model(s): UH-1N
Data as of: January, 1971
DATA BASIS: Bell Model 212
Flight Test

Engine(s): T400-CP-400
Fuel Grade: JP-4/JP-5
Fuel Density: 6.5 Lb/Gal



MODEL: H-46D
ENGINE(s): (1)T58-GE-10
DATA BASIS: ESTIMATED

CONFIGURATION: CLEAN
ROTOR RPM: 100 PERCENT
FUEL GRADE: JP-4/JP-5

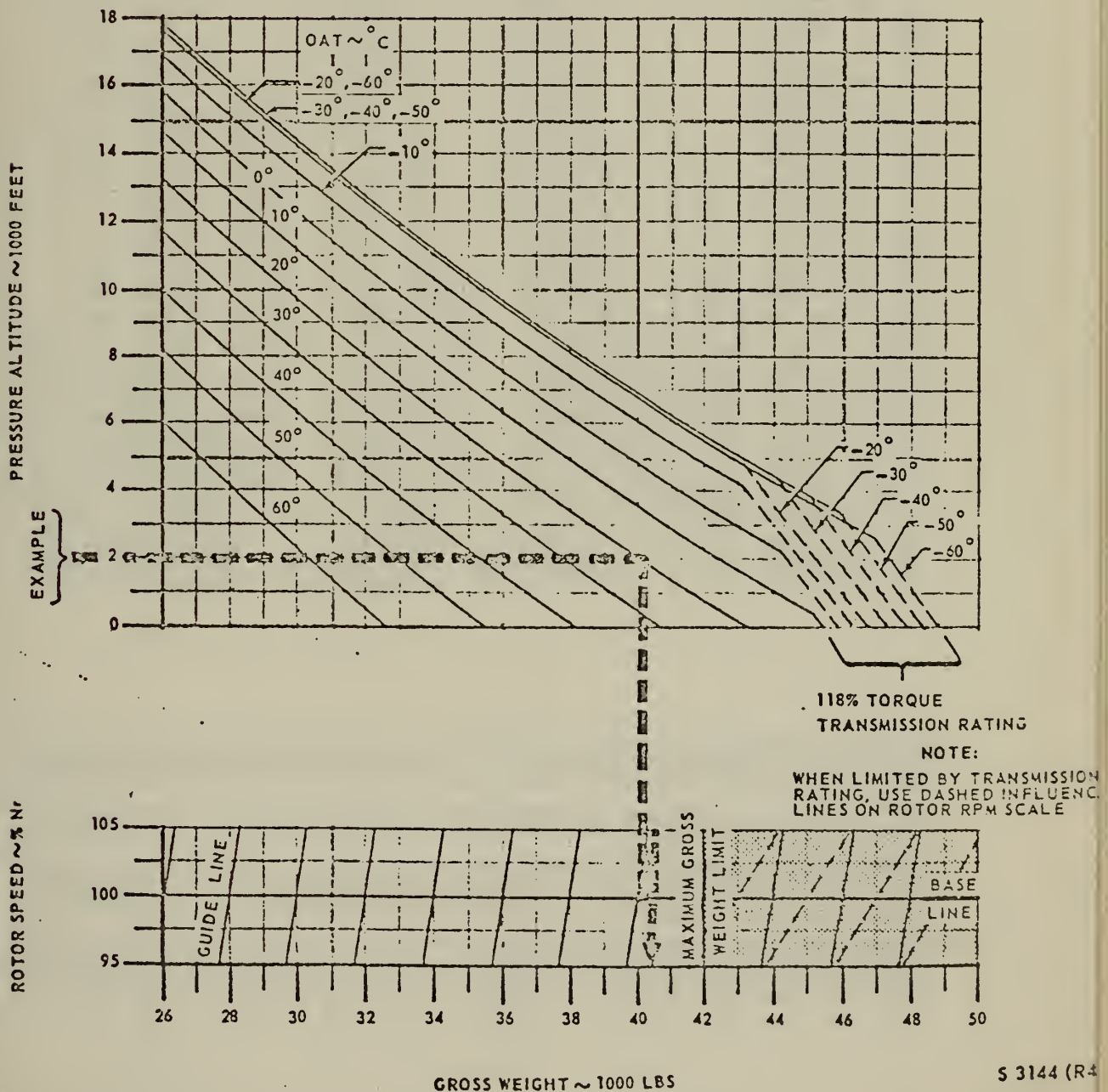


A02-30-10/25

ABILITY TO HOVER OUT OF GROUND EFFECT AT VARIOUS ROTOR SPEEDS MILITARY POWER 699°C T₅

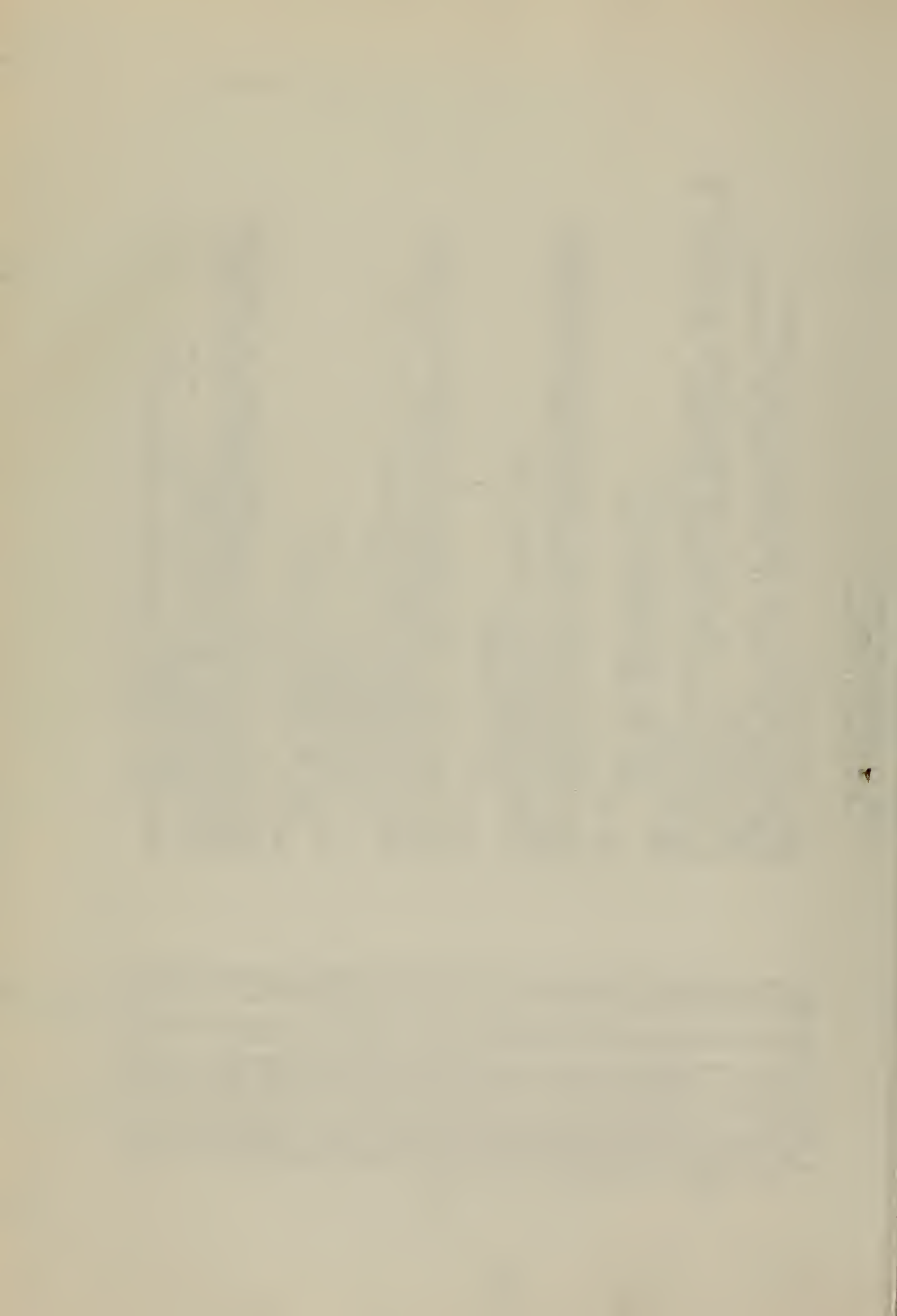
MODEL: CH-53D
DATA AS OF: 1 NOVEMBER 1971
DATA BASIS: FLIGHT TEST

ENGINES: (2) T64-GE-413
FUEL GRADE: JP-4/JP-5
FUEL DENSITY 6.5/6.8 LB/GAL



[illegible]

43



0045	0070	00	046
0046	0074	00	04A
0047	0078	00	04E
0048	0083	00	053
0049	0087	00	057
0050	0092	00	05C
0051	0095	00	05D
0052	0093	00	05D
0053	0093	00	05D
0054	0093	00	05D
0055	0093	00	05D
0056	0093	00	05D
0057	0094	00	05E
0058	0099	00	063
0059	0101	00	065
0060	0105	00	069
0061	0106	00	06A
0062	0106	00	06A
0063	0106	00	06A
0064	0106	00	06A
0065	0106	00	06A
0066	0106	00	06A
0067	0114	00	072
0068	0118	00	076
0069	0122	00	07A
0070	0126	00	07E
0071	0132	00	084
0072	0138	00	08A
0073	0140	00	08C
0074	0146	00	092
0075	0148	00	094
0076	0156	00	09C
0077	0163	00	0A3
0078	0165	00	0A5
0079	0172	00	0AC
0080	0180	00	0B4
0081	0183	00	0B7
0082	0191	00	0BF
0083	0192	00	0C0

```

COMPY: JMS CCMPLEMENT; JLN BACK;
COMPY2: JMS CCMPLEMENT; JMS INIT;
TOPIT: SRC RA; RDO; XCH R5; SRC RC; RDO;
      ADD R5; SRC RE; WRO; LDM O;
      SRC RA; WRO; SRC RC; WRO; XCH R5;
      BBL O;

REM SUBROUTINE SHIFLT SHIFTS THE DESIGNATED RAM
REGISTER
LEFT A DESIGNATED NUMBER OF PLACES. THE NUMBER OF
PLACES TO BE SHIFTED IS PLACED IN CPU REGISTER PAIR C
C THE
RAM REGISTER TC BE SHIFTED SHOULD BE PLACED IN CPU REG
PAIR E F;
SHIFLT: NOP;
      SRC RE; RDM; SRC RC; WRM; INC RD;
      ISZ RF; SHIFLT;
      ISZ RF; WRM; ISZ RD FILLIT; CLC;
      FILLIT: WRM; ISZ RD FILLIT; CLC;
      BBL O;

REM SUBROUTINE MULT TAKES TWO SIX DIGIT NUMBERS,
POSITIVE
OR NEGATIVE MULTIPLIES THEM TOGETHER AND PLACES THE
ANSWER WITH THE CORRECT SIGN IN RAM REG ZERO. THE
NUMBERS TO BE MULTIPLIED ARE ASSUMED TO BE IN RAM REGS
O AND 1;
MULT: FIN; RE "20"; JMS ZERRR; JMS INIT; JMS SETUP;
      LD R2; XCH RB; SRC RA; RDM;
      MULTSTART: XCH R6; JCN 4 DETOUR;
      MULTBEGIN: LD R3; XCH RC; SRC RC; RDM;
      XCH R7; LD R2; ADD R3; CLC; XCH RF; SRC RE;
      LD R7; ADM; DAA; WRM; JCN 10 AROUND;
      TIPPY: LDM 10; XCH R4;
      TIP: INC RF; TCC; SRC RE; ADM; DAA; WRM;
      ISZ R4 TIP;
      LD RF; DAC; DAC; DAC; DAC; XCH RF;
      AROUND: CLC; LCM 5; SUB R3; CLC; INC R3; JCN 4
      FINISHED;
      FINISHED: LDM 0; XCH R3; LD R6; DAC; JCN 4 DETOUR; XCH
      R6;
      ENDING: CLC; LDM 6; SUB R2; CLC; JCN 4 FINI; JUN
      MULTBEGIN;
      DETOUR: LDM 0; XCH R3; INC R2;
      MULTSTART: CLC; LDM 6; SUB R2; CLC; JCN 4 FINI; JUN
      FINI: NOP;
      SRC RE; RDO; CLC;

```


0C84	0155	00	0C3
0C85	02C0	00	0C8
0C86	02C0	00	0CC
0C87	02C6	00	0CE
0C88	02C8	00	0D0
0C89	0212	00	0D4
0C90	0213	00	0D5
0C91	0213	00	0D5
0C92	0213	00	0D5
0C93	0213	00	0D5
0C94	0216	00	0D8
0C95	0219	00	0DB
0C96	0220	00	0DC
0C97	0220	00	0DC
0C98	0220	00	0DC
0099	0220	00	0DC
0100	0220	00	0DC
0101	0220	00	0DC
0102	0220	00	0DC
0103	0220	00	0DC
0104	0224	00	0EO
0105	0238	00	0E4
0106	0232	00	0E8
0107	0234	00	0EA
0108	0239	00	0EF
0109	0243	00	0F3
0110	0245	00	0F5
0111	0246	00	0F6
0112	0247	00	0F7
0113	0247	00	0F7
0114	026C	01	104
0115	0260	01	104
0116	0260	01	104
0117	0260	01	104
0118	0260	01	104
0119	0260	01	104
0120	026C	01	1C4
0121	0260	01	104
0122	0262	01	106
0123	0271	01	10F

```

RAR: JCN 2 AHEAD; JUN HEAP;
AHEAD: FIM RE "20"; JMS CCPLEMENT;
HEAP: FIM RA "20";
FIM RE "00";
JMS ZERRR; JMS MOVE;
BBL 0;

REM SUBROUTINE ZERRR ZERCS OUT THE DESIGNATED
REGISTER. THE REGISTER TO BE ZEROED SHOULD BE PLACED
IN CPU REGISTER PAIR E F;
ZERORR: SRC RE: LDM 0; WRM;
ISZ RF ZERRR; WRC;
BBL 0;

REM SUBROUTINE SHIFRT SHIFTS A DESIGNATED RAM REGISTER
TO THE RIGHT A DESIGNATED NUMBER OF PLACES. THE REGISTER
TO BE SHIFTED SHOULD BE PLACED IN CPU REGISTER E AND
F. THE NUMBER 15 SHCULD BE PLACED IN CPU REGISTER D
AND 15 MINUS THE NUMBER OF PLACES TO BE SHIFTED IN
REGISTER F;
SHIFRT: SRC RE; RDM; SRC RC; WRM;
LD RF; DAC; JCN 10 FILLUP;
XCH RF; LD RD; DAC; XCH RD;
JUN SHIFRT;
FILLUP: CLC; LD RD; DAC; JCN 10 OUT;
XCH RD; LDM 0; SRC RC; WRM;
JUN FILLUP;
OUT: CLC;
BBL 0;

BEGIN 260;

REM SUBROUTINE SWITCH TAKES NUMBERS OUT OF ROM AND
PUTS THEM
INTO RAM REG. IF THE NUMBER IS NEGATIVE IT PLACES THE
TENS
COMPLEMENT IN RAM, THE ASSUMPTICNS OF SWITCH ARE THAT
THE
ADDRESS OF THE ROM CONSTANT TO BE MOVED IS IN CPU REG
PR 0
AND THAT THE RAM REG YCL WANT THE NUMBER IN IS IN CPU
REG
PAIR AB;
SWITCH: LDM 14; XCH RF;
LOOP: JMS LOAC; XCH R2; SRC RA; WRM; INC RB; XCH R3;
SRC RA; WRM;
INC RB;

```


0124	0272	01	110
0125	0273	01	113
0126	0274	01	115
0127	0275	01	11C
0128	0276	01	11E
0129	0277	01	122
0130	0278	01	124
0131	0279	01	127
0132	0280	01	129
0133	0281	01	12B
0134	0282	01	12D
0135	0283	01	130
0136	0284	01	132
0137	0285	01	133
0138	0286	01	133
0139	0287	01	133
0140	0288	01	133
0141	0289	01	133
0142	0290	01	133
0143	0291	01	133
0144	0292	01	133
0145	0293	01	133
0146	0294	01	133
0147	0295	01	133
0148	0296	01	133
0149	0297	01	133
0150	0298	01	133
0151	0299	01	133
0152	0300	01	133
0153	0301	01	133
0154	0302	01	133
0155	0303	01	133
0156	0304	01	133
0157	0305	01	133
0158	0306	01	133
0159	0307	01	133
0160	0308	01	133
0161	0309	01	133
0162	0310	01	133
0163	0311	01	133
0164	0312	01	133
0165	0313	01	133
0166	0314	01	133
0167	0315	01	133

```

ISZ R1 NC11; INC RC; NC11;
ISZ RF LCCP;
LD RB; DAC; XCH RB; SRC RA; REM; WRO; LDM 0;
WRM; RDO; JCN 2 BCITCM;
CLC; RAR; XCH RB;
CLEAR: LDM 3; XCH RA; WRM;
CLEARIT: LDM 0; SRC RA; WRM;
ISZ RB CLEARIT;
JUN DOWN;
BOTTOM: LDM 3; XCH RB;
NEGIT: LDM 9; SRC RA; WRM;
ISZ RB NEGIT;
DOWN:BBL 0;

REM SUBROUTINE SUBTRACT SUBTRACTS THE NUMBER IN RAM
REGISTER
ZERO FROM THE NUMBER IN RAM REGISTER ONE. THE
ANSWER IN THE CORRECT FORM WILL BE PLACED IN RAM
REGISTER
ZERO;
SUBTRACT: JMS INIT; SRC RA; RDO; CLC; RAR;
JCN 2 TCADD;
SBRT: LDM 10; XCH R6;
LDM 0; XCH R7;
CLB;
SUBT: SRC RC; RDM; SRC RA; SBM; JCN 2 OK;
ADD R6; CLC;
OK: CMC; SRC RA; WRM; INC RB; INC RD; ISZ R7 SUBT;
JUN SENDINGS;
TOADD: LDM 0; WRO; FIM RE "00"; JMS CCMPLEMENT;
JMS ADDIT;
SENDINGS: LDM 15; XCH RB; SRC RA; RDM; WRO; CLC;
BBL 0;

REM SUBROUTINE ADDIT ASSUMES TWO NUMBERS ARE IN RAM RE
0 AND 1 THE NUMBER IN REG 0 AND ADDS IT TO THE NUMBER
IT TAKES 1 IS PLACED IN RAM REG 0;
THE RESULT IN REG 1
ADDIT: JMS INIT;
LDM 0; XCH R7;
ADDITION: CLC;
ADDSTART: LD R2; XCH RB;
LD R2; XCH RD;
SRC RA; RCM; DAA;
SRC RC; ACM; WRM;
SRC RA; WRM;
INC R2;

```


0168	0369	01	171
0169	0371	01	173
0170	0377	01	179
0171	0378	01	17A
0172	0378	01	17A
0173	0378	01	17A
0174	0378	01	17A
0175	0378	01	17A
0176	0381	01	17D
0177	0386	01	182
0178	0389	01	185
0179	0392	01	188
0180	0392	01	188
0181	0392	01	188
0182	0392	01	18A
0183	0394	01	18C
0184	0396	01	18C
0185	0400	01	190
0186	0404	01	194
0187	0408	01	198
0188	0412	01	19C
0189	0414	01	19E
0190	0414	01	19E
0191	0414	01	19E
0192	0414	01	19E
0193	0418	01	1A2
0194	0425	01	1A5
0195	0425	01	1A9
0196	0425	01	1A9
0197	0427	01	1AB
0198	0428	01	1AC
0199	0431	01	1AF
0200	0434	01	1B2
0201	0436	01	1B4
0202	0440	01	1B8
0203	0443	01	1BB
0204	0445	01	1BD
0205	0445	01	1C1
0206	0452	01	1C4
0207	0453	01	1C5
0208	0453	01	1C5
0209	0453	01	1C5
0210	0453	01	1CB
0211	0459	01	1CB
0212	0459	01	1CB

```

ISZ R7 ADDSTART;
LDM 15; XCH RB; SRC RA; RDM; WRO; CLC;
DOME: BBL 0;

REM SUBROUTINE ASCOL LOCKS AT THE THOUSANDS COLUMN AND
THE
HUNDREDS COLUMN OF THE AC WIND GROSS WEIGHT AND SETS UP
THE CPU FOR CCLNUMAS;
ASCOL: LDM 4; SRC R4; SBM;
JCN 2 NEXT; INC R4; RE; JUN NEXT;
NEXT: INC R5; SRC R4; RDM;
XCH R9; CLC; BBL 0;

REM SUBROUTINE CCLNUMAS DETERMINES THE CORRECT AIRSPEED
COLUMN TO ENTER THE DATA WITH;
CCLNUMAS: LDM 2; XCH R6;
CLB; XCH R7;
LOOPS: LD R8; DAC; JCN 1C COLVALA;
XCH R8; LD R6; CLC; ADD R7;
XCH R7; CLB; JUN LCCPS;
COLVALA: LD R7; IAC; ADD R8; XCH R8;
CLC; BBL 0;

REM SUBROUTINE TESTPA DETERMINES WHETHER THE ENTERED
ALTITUDE
IS IN THE RANGE 0-20,000 FEET;
TESTPA: JCN 2 BYIT; SRC R8;
SBM; JCN 2 BYIT; JUN ERRCR; BYIT: CLC; BBL 0;

REM SUBROUTINE INITC LCADS THE PROPER CONSTANTS INTO
THE CPU BEFORE
THE GROSS WEIGHT IS CALCULATED;
INITC: JMS LLOAD;
LD R2;
XCH R4; LD R3; XCH R5;
ISZ R1 NOINC; INC R0;
NOINC: JMS LCAL;
LD R2; XCH R6; LD R3; XCH R7;
ISZ R1 NOINC1; INC R0;
NOINC1: JMS LCAD;
LD R2; XCH R8; LD R3; XCH RA;
ISZ R1 NOINC2; INC R0;
NOINC2: BBL 0;

REM SUBROUTINE TEMPCCL SETS UP THE CPU WITH THE
PROPER NUMBERS FOR ENTRY INTO CCLNUM;
TEMPCCL: SRC R4; RDM; JCN 4 UPPER; JUN ERRCR;
REM DETERMINE COLUMN OF TABLE;
UPPER: CLB; XCH R5; SRC R4; RDM;

```



```

0213 0463 01 1CF      INC R5; JCN 4 TENS; REM IF ACC N.F. 0 THEN
0214 0466 01 1D2      CONTINUE;
0215 0469 01 1D5      CLB; IAC; XCH R9;
0216 0471 01 1D7      TENS: SRC R4; RDM; REM PUT CHAR TO BE CHECKED IN R9 ;
0217 0475 01 1DB      ADD R9; XCF R9; CLC; BBL 0;
0218 0475 01 1DB      REM SUBROUTINE COLNUM DETERMINES THE PROPER COLUMN
0219 0475 01 1DB      NUMBER
0220 0475 01 1DB      TO ENTER THE TEMPERATURE DATA WITH;
0221 0475 01 1DF      CCLNUM: CLC; LC R6; ACC RA; XCH R6;
0222 0483 01 1E3      LD R6; SUB R9; JCN 2 COLVAL; REM IF THE BORROW
0223 0483 01 1E3      IS NOT USED THEN COLUMN NUMBER IS IN R8;
0224 0485 01 1E5      ISZ R8 COLNUM; REM IF COUNTER STILL IN RANGE
0225 0487 01 1E7      CONTINUE;
0226 0489 01 1E9      JUN ERROR;
0227 0491 01 1EB      COLVAL: CLC; LD R8; REM SET R8 TO COLUMN NUMBER AND
0228 0492 01 1EC      RETURN;
0229 0492 01 1EC      SUB R7; XCH R8;
0230 0492 01 1EC      BBL 0;
0231 0492 01 1EC      REM SUBROUTINE INIT CLEARS THE ACCUMULATOR AND THE
0232 0492 01 1EC      CARRY
0233 0492 01 1EC      THEN PUTS ZERGES INTO CPL REG 2 3 4 5 6 7 8 9 AND A
0234 0492 01 1EC      IN REG PAIR AB AND A '10' IN REG PAIR CD AND A '20' IN
0235 0492 01 1EC      IN REG PAIR EF;
0236 0492 01 1EC      INIT: CLB;
0237 0492 01 1EC      FIM R2 "CC";
0238 0492 01 1EC      FIM R4 "00";
0239 0492 01 1EC      FIM R6 "00";
0240 0492 01 1EC      FIM R8 "00";
0241 0492 01 1EC      FIM RA "00";
0242 0492 01 1EC      FIM RC "10";
0243 0492 01 1EC      FIM RE "20";
0244 0492 01 1EC      BBL 0;
0245 0492 01 1EC      BEGIN 512;
0246 0492 01 1EC      REM SUBROUTINE LOADROBAS INITIALIZES THE CPU REGISTERS
0247 0492 01 1EC      WITH CONSTANTS STORED IN FRONT OF THE CONSTANT TABLES.
0248 0492 01 1EC      THE CONSTANTS; TABLE BASE ADDRESS AND NUMBER OF ROWS
0249 0492 01 1EC      PER
0250 0492 01 1EC      CCLNUM ARE USED TO COMPUTE TABLE ENTRY ADDRESS. ;
0251 0492 01 1EC      LOADROBAS: JMS LOAD; REM PUT NO. OF ROWS/CCCL IN R7;
0252 0492 01 1EC      LD R3; XCH R7; CLC; LDM 1;
0253 0492 01 1EC      ADD R1; XCH R1; TCC; ADD R0;
0254 0492 01 1EC      XCH R0; JMS LCAC; LD R2; XCH R0;

```



```

RE "10"; JMS ZERRR;
FIM RA "13";
FIM SRC LCM 1; WRM;
FIM SUBTRACT;
FIM RA "10";
JMS SWITCH;
JMS MULT;
FIM RA "50"; FIM RE "10";
JMS ZERRR; JMS MOVE;
JMS MULT;
FIM RC "00"; FIM RE "03";
JMS SHIFLT;
FIM RA "10";
JMS SWITCH;
FIM RC "1F"; FIM RE "1D";
JMS SHIFRT;
JMS ADDIT;
FIM RA "00"; FIM RE "7C";
JMS ZERRR; JMS MOVE;
FIM RA "10";
JMS SWITCH;
FIM RA "40"; FIM RE "00";
JMS ZERRR; JMS MOVE;
JMS MULT;
FIM RA "70"; FIM RE "10";
JMS ZERRR; JMS MOVE;
JMS ADDIT;
FIM RA "00"; SRC RA; RCL;
FIM

TEST FOR FLAG SET TC INDICATE SECOND ITERATION. ;
REM JCN 12 XINIT;

TEST COMPUTED GROSS WEIGHT FCR POSSIBLE REGION 3.
FIM RA "00"; FIM RE "50"; JMS ZERRR; JMS MCVE;
FIM RA "10"; JMS SWITCH; FIM RA "50";
FIM RE "00"; JMS ZERRR; JMS MOVE;
JMS MULT;
JMS SWITCH;
FIM RA "10"; FIM RE "1D";
FIM RC "1F";
JMS SHIFRT;
JMS ADDIT;
FIM RA "00"; FIM RE "1C"; JMS ZERRR; JMS MCVE;
FIM RA "90"; FIM RE "00"; JMS ZERRR; JMS MOVE;
JMS SUBTRACT; SRC RA; RCM; WRL;
FIM RA "0F";

TEST FCR GROSS WEIGHT IN REGION 1. ;
REM JCN 12 START2;

```


0396	1110	04	456	WRM; JMS MLLT;
0397	1113	04	459	FIM RA "10"; JMS SWITCH;
0398	1117	04	450	JMS MULT; FIM RC "00"; FIM RE "02"; JMS SHIFLT;
0399	1125	04	465	FIM RA "90"; FIM RE "10";
0400	1129	04	465	JMS ZERRR; JMS MOVE;
0401	1133	04	460	JMS ADDIT;
0402	1135	04	465	FIM RA "00"; FIM RE "SC"; JMS MOVE;
0403	1141	04	475	ASZERC: NOP;
0404	1142	04	475	REM MOVE COMPUTED GROSS WEIGHT TO DISPLAY REGISTER. ;
0405	1142	04	476	FIM RA "90"; FIM RE "FC"; JMS MOVE;
0406	1142	04	476	FIM RA "F4";
0407	1150	04	470	FILLMUP: SRC RE; LDM "F"; WRM;
0408	1153	04	481	ISZ RF FILLMUP;
0409	1155	04	485	JUN CHECKNOSTROBE;
0410	1157	04	485	BEGIN 1280;
0411	1157	04	485	REM SUBROUTINE DELAY IS CALLED TO COMPENSATE FOR
0412	1157	04	485	MECHANICAL INCONSISTANCIES OF KEYBOARD INPUT. ;
0413	1280	05	500	DELAY: FIM RC "00";
0414	1280	05	500	ITDELAY: ISZ R7 ITDELAY;
0415	1280	05	500	ITDELAY: ISZ R6 ITDELAY;
0416	1282	05	502	BBL 0;
0417	1284	05	504	REM INPUT OUTPUT ROUTINES FOLLCK. ;
0418	1286	05	506	CHECK: SRC RC; RDM; IAC; CLC;
0419	1287	05	507	JCN ZAC OVERALL;
0420	1287	05	507	DAC; CLC; SRC RE; WRM; INC RF;
0421	1287	05	507	ISZ RC CHECK;
0422	1291	05	508	OVERALL: BBL 0;
0423	1293	05	512	CHECKSTROBE: FIM RE "FO"; JMS "600";
0424	1294	05	512	FIM RC "10"; SRC RC; RCR; RAL;
0425	1296	05	514	JCN 10 CHECKSTROBE; JMS DELAY; FIM RC "00"; SRC RC;
0426	1300	05	514	RDR; XCH RC; RDR; RAL;
0427	1301	05	515	INC RC; SRC RC; RDR; RAL;
0428	1305	05	519	JCN 10 CHECKSTROBE;
0429	1305	05	519	FIM RC KEYTAB; RAL; JCN 10 NUMBERSIDE;
0430	1323	05	527	INC RC; XCH RC; RDR; RAL;
0431	1325	05	528	JCN 10 CHECKSTROBE;
0432	1330	05	532	FIM RC KEYTAB; RAL; JCN 10 NUMBERSIDE;
0433	1332	05	534	NUMBERSIDE: IDE; LD RC; RD;
0434	1333	05	535	CLC; ADD RD; JCN 10 DC1; INC RO;
0435	1338	05	535	DC1: CLC; ADD RD; JCN 10 XCH RI;
0436	1341	05	535	DC2: LD RD; JCN 10 INC RO;
0437	1344	05	540	JCN 10 DC2; JCN 10 INC RO;
0438	1346	05	542	CHECKNOSTROBE: FIM RE "FC"; JMS "600";
0439	1350	05	546	FIM RC "10"; SRC RC;
0440	1353	05	549	RDR; RAL; JCN 2 CHECKNOSTROBE;
0441	1357	05	549	JUN CHECKSTROBE;
0442	1357	05	549	

0443	1359	05	54F	KEYTAB: JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0444	1361	05	551	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0445	1367	05	557	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0446	1373	05	55D	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0447	1379	05	563	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0448	1385	05	569	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0449	1387	05	56B	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0450	1391	05	56F	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0451	1393	05	571	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0452	1395	05	573	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0453	1397	05	575	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0454	1403	05	57B	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0455	1405	05	57D	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0456	1407	05	57F	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0457	1409	05	581	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0458	1415	05	587	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0459	1417	05	589	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0460	1419	05	58B	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0461	1423	05	58F	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0462	1423	05	58F	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0463	1423	05	58F	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0464	1427	05	593	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0465	1431	05	597	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0466	1437	05	59D	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0467	1441	05	5A1	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0468	1447	05	5A7	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0469	1449	05	5A9	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0470	1455	05	5AF	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0471	1461	05	5B5	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0472	1465	05	5B9	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0473	1467	05	5BB	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0474	1469	05	5BC	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0475	1475	05	5C3	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0476	1477	05	5C5	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0477	1481	05	5C9	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0478	1487	05	5CF	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0479	1489	05	5D1	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0480	1493	05	5D5	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0481	1501	05	5DQ	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0482	1505	05	5E5	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0483	1511	05	5E7	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0484	1518	05	5EE	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0485	1520	05	5FC	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;
0486	1523	05	5F3	JUN	CHECKNCSTRCBE;	NUMBER;	JUN	NUMBER;


```

CON "0400999158816910151C0600210C5981641C0330";
CON "9991611089915691221C0400200C588115105220";
CON "080029919781601C891C0C006010000056916C10";
CON "08002000039811010252C0600080CC11007CC900";
CON "0210070000100310C8CC0010051CC90C01100520";
CON "0010521CC0C0010CC00000";
ERRGR: FIM RE; "FOM"; 9; WRN;
ERR1: SRC RE; LDM; 9;
JUN CHECKNOSTROBE;
LOAD: FIM R2;
LI: BBL 0;
FINX: NOP;
END

```

```

05333 07 73C
05334 07 750
05335 07 764
05336 07 778
05337 07 78C
05338 07 7A0
05339 07 7AC
05340 07 7AE
05441 07 7B1
05442 07 7B3
05443 07 7B5
05444 07 7B6
05445 07 7B7
05446 07 7B8

```


V. LIST OF REFERENCES

1. U.S. ARMY AVIATION SYSTEMS TEST ACTIVITY, Engineering Flight Evaluation of Helicopter Lift Margin System Test Plan, by F. Dominick, April 1973.
2. ELLIOTT ELECTRONICS, Hermes Helicopter Energy and Rotor Management System, p. 2-17.
3. INTEL CORPORATION, MCS-4 Micro Computer Users Manual, Revision 3, July 1972.
4. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-230HMA-1, NATOPS Flight Manual Navy Model CH-53 A/D Helicopters, p. 11-23, 1 November 1971.
5. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-110HCE-1, NATOPS Flight Manual UH-IN Helicopter, p. 11-20, January 1971.
6. NAVAL AIR SYSTEMS COMMAND NAVAIR 01-250HDA-1, NATOPS Flight Manual H46-D Helicopter, p. 11-41, 6 May 1965.
7. NAVAL AIR SYSTEMS COMMAND NAVAIR 01 110HCA-1, NATOPS Flight Manual UH/TH-IL, HHIK Helicopters, p. 11-76, August 1969.
8. NAVAL AIR TEST CENTER USNTPS-FTM NO. 101, Naval Test Pilot School Flight Test Manual Helicopter Stability and Control, 10 June 1968.
9. RICHARDS, R.B., Principles of Helicopter Performance, Naval Air Test Center, USNTPS-T-No. 1, 8 March 1968.

INITIAL DISTRIBUTION LIST

	No. of Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Chairman Computer Science Group, Code 72 Naval Postgraduate School Monterey, California 93940	1
4. Professor G.A. Kildall, Code 53 Kd Naval Postgraduate School Monterey, California 93940	1
5. Lieutenant (j.g.) R.H. Brubaker, Jr., Code 53Bh Naval Postgraduate School Monterey, California 93940	1
6. Capt. E.E. Eloee 14521 Apt. A Carfax Tustin, California 92680	1
7. Lt. R.T. Scott, Jr. 1428 Carissa Ct., Chula Vista, California 92011	1
8. Mr. G. Flohill, Code 461 Office of Naval Research Arlington, Virginia 22217	1
9. Commander US Army Aviation System Test Activity Attn: SAVTE-M (Mr. Farrell) Edwards AFB, California 93523	1
10. Major Richard L. Phillips, USMC Sub Unit #1, MCTSSA Marine Corps Air Station (H) Santa Ana, California 92075	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Naval Postgraduate School Monterey, California 93940		Unclassified	
3. REPORT TITLE		2b. GROUP	
A Helicopter Flight Performance System Using an LSI Microprocessor.			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)			
Master's Thesis; (June 1973)			
5. AUTHOR(S) (First name, middle initial, last name)			
Edwin Eugene Eloie and Richard Tazewell Scott, Jr.			
6. REPORT DATE		7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1973		59	9
8. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
9. PROJECT NO.			
10. DISTRIBUTION STATEMENT		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT			
<p>This thesis presents the development of a helicopter gross weight calculator. Required qualities for aircraft system components such as durability, reliability, and low weight are met using an LSI micro-processor. The prototype system which was developed weighs approximately four pounds and has approximate dimensions of 7" x 5" x ½". The cost estimate for the system is less than \$350. The calculator solution is based on the solution technique currently being used by Naval Aviators which is obtained from nomographs in the aircraft NATOPS manual. Minor modifications are required to make this system applicable to different helicopter types. A listing of the calculator program and a discussion of the prototype's operation are included.</p>			

Thesis

E426 Elo

145272

c.1

A helicopter flight
performance system using
an LSI microprocessor.

12 MAY 75

1 AUG 75

20 JUN 76

25 FEB 86

23019

23618

24450

31409

Thesis

E426 Elo

145272

c.1

A helicopter flight
performance system using
an LSI microprocessor.

thesE426

A helicopter flight performance system u



3 2768 002 06985 8

DUDLEY KNOX LIBRARY